



Dynamic Tool Orchestration for Iterative Visual Reasoning

Poster: TBD



ICLR





- **Motivation**
- **Contribution**
- **Methods**
- **Experiments**
- **Example**

Motivation

- Humans often rely on external tools to solve the complex reasoning problems that go beyond what they can handle through internal thinking alone. So models should also:
 - Know **how** to use a tool.
 - Know **what** tool to use and adjust them under different tasks.
 - Can perform with tools even the tool definitions have been **changed**.
- Existing tool-augmented models fall in:
 - Single-tool** trajectories or single-turn loops.
 - Cannot generalize to new tasks and new tool definitions.

Contribution

- A **unified recipe** for tool-aware reasoning:
 - Multi-turn tool planning data, RL-based tool interaction, and an adaptive learning method for strong generalization.
- Self-adaptive tool usage:
 - Automatically selects useful tools, avoids irrelevant ones, and adjusts usage frequency while generalizing to **unseen tools** and **new tasks**.

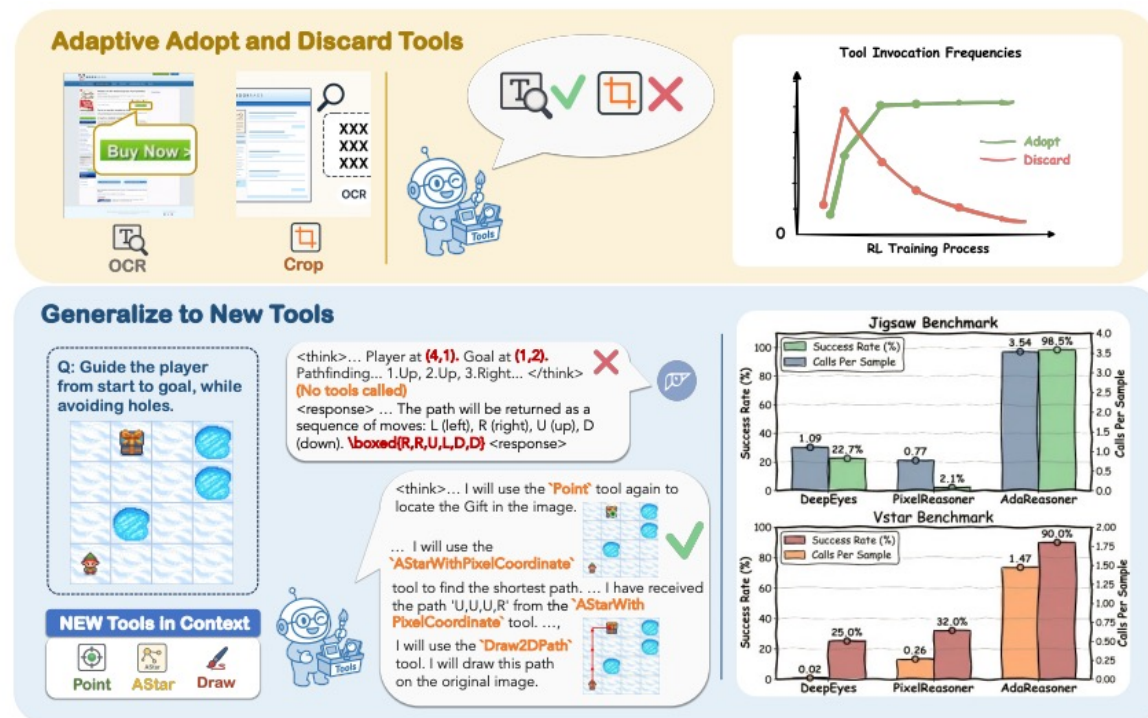


Figure 1. AdaReasoner demonstrates adaptive tool-use behavior and strong generalizability across diverse tasks.

- SOTA performance:
 - Surpasses its base models and rivals or exceeds leading **proprietary models** (e.g., GPT-5, Claude Sonnet 4) on structured reasoning tasks.

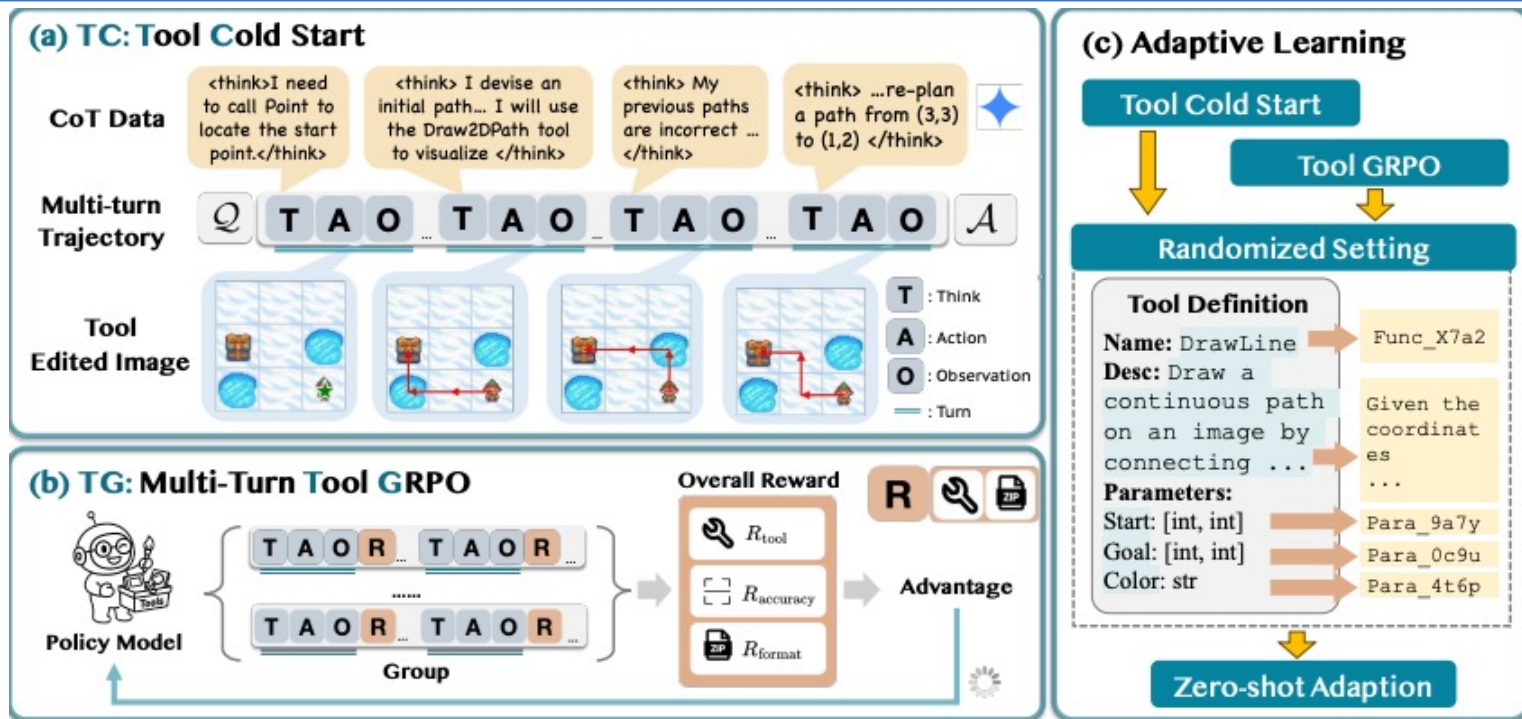


Figure 2. An overview of the AdaReasoner method. (a) the Tool Cold Start (TC) phase (b) the Tool GRPO (TG) (c) the Adaptive Learning Method

Tool Cold Start (TC)

- We design a data curation framework and curate a batch of high quality Tool Cold-Start data.
- Data Curation:
 - Design the abstract trajectory blueprints. (Including reflection, backtracking & explicit tool failure)
 - Use tool server to supply the trajectory with tool execution results

- Use Gemini to supply the trajectory with CoT data

Tool GRPO (TG)

- **Multi-turn Reward Accumulation:** Calculate Tool reward, Format reward from every turn to ensure a stable tool-calling
- **Adaptive Reward for Encouraging Tool Use:** We design an adaptive tool reward. When the answer is correct, the model gets full reward. When the answer is not correct, tool reward is applied to encourage a tool-calling behavior under uncertainty.

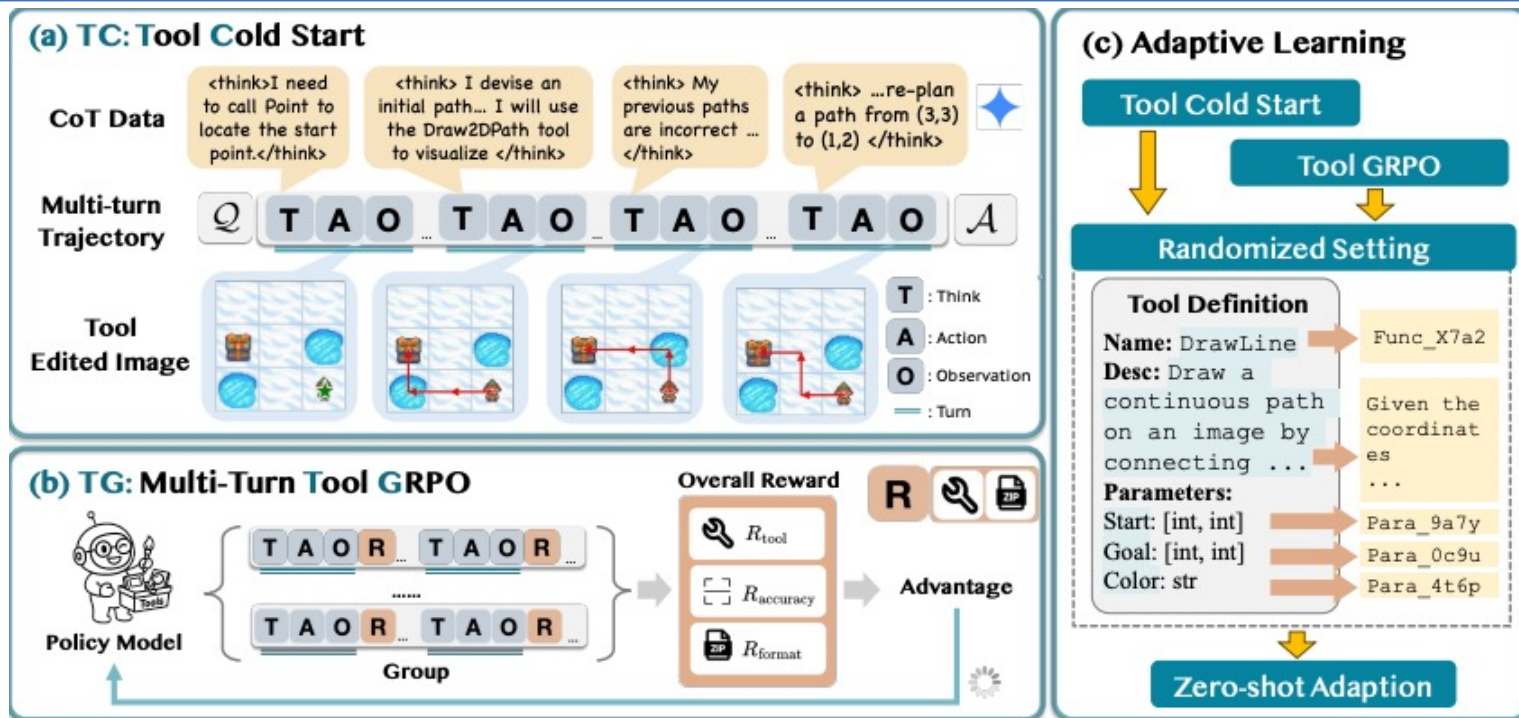


Figure 2. An overview of the AdaReasoner method. (a) the Tool Cold Start (TC) phase (b) the Tool GRPO (TG) (c) the Adaptive Learning Method

Adaptive Learning

- **Token-Level Randomization for Identifiers:** We hypothesize that a robust tool planner should not depend on semantically meaningful identifiers (e.g., relying on the word “Calculator” to know it performs math). We thus apply a replacement policy to all functional identifiers, including tool names and argument names.
- **Semantic-Level Paraphrasing for Descriptions:** We leverage Gemini 2.5 Flash to rephrase the original descriptions. The objective is to alter the syntactic structure and lexical choices while strictly maintaining the original functional meaning.

Model	VSP0	VSP	Jigsaw	BLINK-J	GUIChat	WebMMU [†]
Qwen2.5-VL 3B	26.53	26.73	39.80	48.67	45.11	55.89
+ Direct SFT	38.15	38.82	42.60	53.33	55.51	61.38
+ Direct GRPO	24.55	32.73	42.70	52.67	52.49	56.30
+ TC (cold-start)	47.01	51.09	66.00	70.00	45.32	44.72
+ TG (tool-GRPO)	29.10	35.09	43.00	47.33	89.60	72.15
+ TC + TG	<u>84.73</u>	<u>94.73</u>	<u>94.80</u>	<u>88.67</u>	85.45	81.71
Δ vs. base	+58.20	+68.00	+55.00	+40.00	+40.34	+25.82
Qwen2.5-VL 7B	25.39	28.09	45.70	52.67	59.46	67.48
+ Direct SFT	42.18	46.64	<u>86.40</u>	<u>88.00</u>	62.68	65.65
+ Direct GRPO	28.38	30.18	64.90	80.00	67.67	<u>83.54</u>
+ TC (cold-start)	<u>61.69</u>	<u>64.91</u>	84.20	83.33	61.85	64.63
+ TG (tool-GRPO)	59.70	73.18	72.30	80.67	92.52	88.62
+ TC + TG	85.09	97.64	96.60	96.00	<u>88.57</u>	82.32
Δ vs. base	+59.70	+69.55	+50.90	+43.33	+29.11	+14.84

Table 1. Performance improvements of the base model under different training methods.

Visual Tools Help Bridging The Capability Gap

We first fine-tune the model on **each individual task** and systematically evaluate the contribution of TC and TG.

- **Visual Tools Bring Stable Improvements:** Our TC+TG demonstrates an average gain of **+38.66%** on the 7B model. This tool-augmented approach transforms tasks like VSP from an under-optimized baseline (~31.64%) to near-perfect execution (97.64%).
- **Visual Tools Help Overcome Scale-Based Limitations:** while the baseline performance of 3B and 7B models is disparate and low, our tool-augmented versions both achieve near-perfect accuracy (94.7% and 97.6%). This suggests that the primary performance bottleneck has **shifted** from the model’s scale to the **intrinsic quality of the tools** it employs.

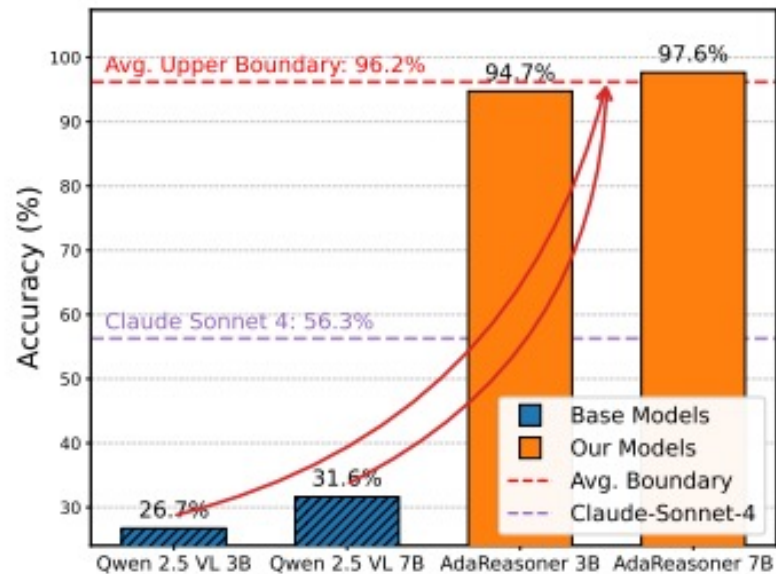


Figure 3. Improvements brought by TC+TG

Stage	Reflection	A*	VSP			VSPO			A* Statistics	
			Nav	Verify	Overall	Nav	Verify	Overall	CPS	Succ.
TC + TG	✗	RL	96.33	<u>99.20</u>	97.64	73.44	<u>98.70</u>	85.09	0.56	100.0
TC + TG	✓	-	<u>84.33</u>	99.80	<u>91.36</u>	<u>63.89</u>	99.61	<u>80.36</u>	0.00	0.0
TC + TG	✓	Inf	55.17	84.60	68.55	57.22	99.61	76.77	<u>0.68</u>	16.9
TC + TG	✗	Inf	62.33	80.00	70.36	43.78	88.70	64.49	0.52	<u>94.5</u>
TC Only	✗	-	41.00	93.60	64.91	31.58	94.01	61.69	0.00	0.0
TC + TG	✗	-	44.83	94.20	67.27	27.67	94.81	58.62	0.00	0.0
TC Only	✗	Inf	46.00	79.40	61.18	32.11	81.43	54.85	0.49	85.2

Table 3. The model can learn to use new tools during inference time.

AdaReasoner Can Use New Tools during Inference Time

Key Idea

- We test whether AdaReasoner can leverage a new tool (A*) that is never seen during TC training.
- A* is only introduced at inference time.

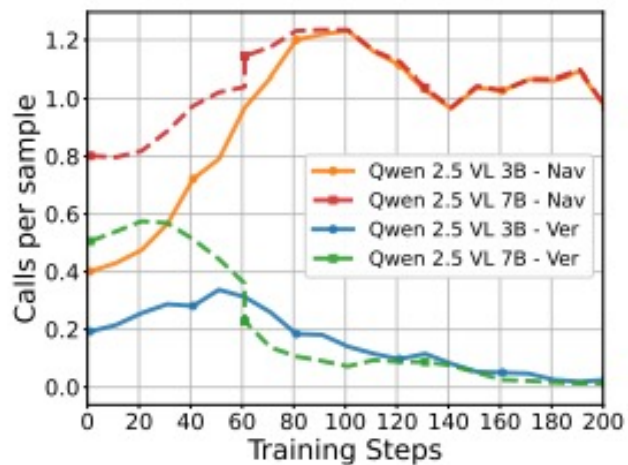
Findings

- Adding the A* tool only during inference provides a major performance boost.
- VSP Navigation improves from 44.83 → 62.33 (TC+TG model without reflection).

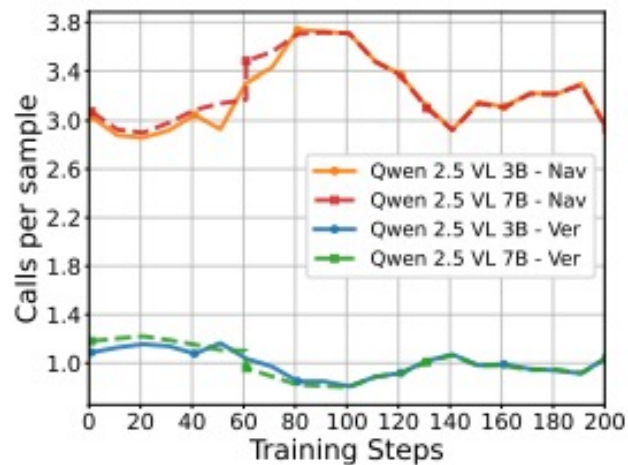
Limitations

Although the model can adaptively invoke new tools at inference,

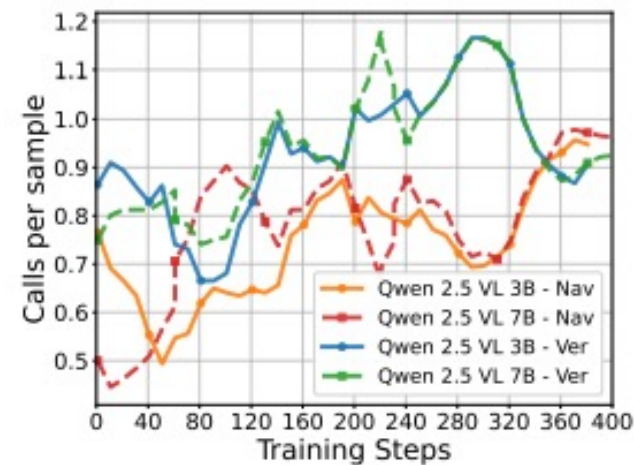
- its tool-adaptation is unstable
- RL (TG) **is still required** to stabilize new-tool usage



(a) ASTAR



(b) POINT



(c) DRAW2DPATH

Figure 4. The model can learn adaptive tool-calling during TG (RL).

AdaReasoner can learn adaptive tool using

We further apply RL to stabilize the model’s tool-calling behavior, and observe that the model develops new forms of adaptive tool usage during the RL stage.

- Learning to Adopt Beneficial Tools:** For the Path Navigation task (warm-colored curves), the model’s invocation frequency for ASTAR progressively increases, stabilizing at a high rate of over 1.0 call per sample.
- Learning to Discard Irrelevant Tools:** For the Verification task. The model initially explores using the A* tool but, receiving no reward for doing so, gradually learns to suppress its usage, with the invocation frequency decaying towards zero.
- Learning to Modulate Tool-Use Frequency:** Beyond the binary choice of adopting or discarding a tool, the model exhibits a more nuanced adaptive behavior.



Model	VSP0	VSP	Jigsaw	BLINK-J	GUIChat	WebMMU [†]
Qwen2.5-VL 7B	25.39	28.09	45.70	52.67	68.09	67.48
+ TC	27.66	29.36	<u>83.50</u>	<u>82.00</u>	36.59	41.87
+ Rnd TC	26.71	30.36	77.90	80.00	35.55	42.89
+ TG	24.01	30.73	66.10	82.00	<u>82.54</u>	67.68
+ TC + TG	24.43	28.00	59.20	78.00	80.35	59.76
+ Rnd TC + TG	<u>57.78</u>	<u>47.27</u>	90.60	91.33	82.74	<u>69.11</u>
+ Rnd TC + Rnd TG (Ours)	69.94	78.91	83.50	80.67	80.87	70.93
Δ vs. base	+44.55	+50.82	+37.80	+28.00	+12.79	+3.46

Table 4. The tool-calling ability can generalize to other tasks.

AdaReasoner: Generalized Tool-Use Capability

Why Generalization Matters

- Applying TC + TG alone is not enough for robust generalization
- Models may still struggle when facing unseen tools or new task distributions

1. Generalization to New Tasks

- TC stage uses only Jigsaw data
- All VSP and WebQA training data is **withheld** during TC
- TG stage uses all data from the three tasks
- **Results:** The tool-planning skills learned from Jigsaw can transfer to unseen tasks like VSP and WebQA.

2. Generalization to New Tools

- During evaluation, the tool definitions differ completely from those used in training
- Tool functionality remains the same, but the interfaces are unseen
- **Results:** The model can handle new tool interfaces instead of memorizing tool formats to complete new tasks.

Model	VSPO	VSP	Jigsaw	BLINK-J	GUIChat	WebMMU	HRBench	V*	Avg.
<i>Closed-Source Models</i>									
Gemini 2.5 flash	40.12	53.55	67.20	65.33	83.05	66.26	78.25	68.59	65.54
GPT 5	34.25	55.64	80.10	73.33	71.41	80.49	74.38	74.87	68.56
Claude 4 sonnet	51.56	56.27	58.60	65.33	93.14	83.54	60.62	59.69	66.09
<i>Open-Source Models</i>									
Qwen 2.5 VL 3B	26.53	26.73	39.80	48.67	46.26	54.47	53.00	43.98	42.93
Qwen 2.5 VL 7B	25.39	28.09	45.70	52.67	68.09	67.48	63.62	63.35	51.80
Qwen 2.5 VL 32B	28.56	33.91	59.50	64.67	85.21	85.98	70.12	72.25	62.53
Qwen 2.5 VL 72B	33.41	39.09	70.10	71.33	88.01	91.06	73.00	80.10	68.76
InternVL3 78B	28.14	35.09	52.80	60.00	79.83	71.34	75.12	81.15	60.93
<i>Tool-Planning Models</i>									
Qwen 2.5 VL 7B + Tools	28.98	30.45	45.00	56.00	56.76	69.51	63.75	54.97	50.93
Qwen 2.5 VL 72B + Tools	39.64	45.00	61.50	65.33	77.13	<u>76.83</u>	67.12	65.97	62.82
GPT 5 + Tools	<u>52.75</u>	<u>71.36</u>	<u>84.50</u>	<u>76.00</u>	<u>76.51</u>	88.82	78.50	<u>70.16</u>	<u>74.83</u>
DeepEyes*	10.18	12.18	44.80	50.67	65.90	72.76	67.00	67.54	48.88
PixelReasoner*	21.38	24.55	52.20	59.33	72.45	69.51	52.12	53.40	50.87
AdaReasoner 7B	71.08	78.36	88.60	88.00	73.91	72.15	<u>69.12</u>	70.68	76.49
Δ vs. base	+45.69	+50.27	+42.90	+35.33	+5.82	+4.67	+5.50	+7.33	+24.72

Table 5. Main results of AdaReasoner.

Model	Jigsaw				Vstar				Orig.	Acc.
	# Turns	CPS	Succ(%)	Acc.	# Turns	CPS	Succ(%)	Acc.		
DeepEyes	1.51	1.09	22.71	44.80	1.01	0.02	25.00	67.54	90.10	
PixelReasoner	1.13	0.77	2.09	52.20	1.19	0.26	32.00	53.40	84.30	
GPT5 + Tools	1.05	0.00	0.00	<u>84.50</u>	1.24	0.24	89.13	<u>70.16</u>	–	
Qwen 2.5 VL 7B + Tools	3.40	<u>2.06</u>	64.41	45.00	<u>1.91</u>	0.18	<u>94.29</u>	54.97	–	
Qwen 2.5 VL 72B + Tools	<u>4.24</u>	0.57	<u>90.40</u>	61.50	1.72	<u>0.27</u>	100.00	65.97	–	
AdaReasoner 7B	4.49	3.54	98.50	88.60	2.35	1.47	90.04	70.68	–	

Table 6. Tool-invoking statistics of AdaReasoner.

Main Results

We select the data from VSP, Jigsaw, and WebQA as our cold start data and VSP, Jigsaw, WebQA, and Visual Search as our RL data.

To enhance the model’s generalizable tool-use capabilities, we applied the Adaptive learning method to get our final model.

Overall Performance Across Tasks:

Our model achieves consistent and substantial **performance improvements** across both visual reasoning tasks and general multimodal tasks.

Generalizable Tool Using:

AdaReasoner generalizes across **new tools and new tasks**, showing strong tool usage even when tool definitions are entirely changed at evaluation time. It maintains **high CPS and near-perfect execution success**, while baselines either barely use tools or fail to execute them reliably under domain shifts.

Thanks!

Poster: TBD



Model	Perception	VSP-Verify Reasoning Accuracy		
	Point Loc. Acc. \uparrow	Base	w/ Line	w/ Point
Qwen 2.5 VL 3B	2.47	50.91	57.92 (+7.01)	49.09 (-1.82)
Qwen 2.5 VL 7B	47.01	50.85	57.68 (+6.83)	57.87 (+7.02)
Qwen 2.5 VL 32B	6.54	53.12	61.31 (+8.19)	87.87 (+34.75)
Qwen 2.5 VL 72B	50.00	52.34	61.57 (+9.23)	87.53 (+35.19)
Our POINT Tool	100.0	–	–	–

Table 2. Our perceptual tools and verifiers can bring positive effects to the model.

Why Visual Tools Can Help?

- **Perception Tools Help MLLMs to See:** In VSP-verification, our expert POINT tool achieves perfect localization accuracy (100.0% vs. \sim 50.0% for baselines), and providing its coordinate output as context boosts the downstream zero-shot reasoning performance by an average of +18.79 points.
- **Manipulating Tools Help MLLMs to Verify:** Under a zero-shot context-appending setting, the DRAWLINE does help improve the judge accuracy of the model, yielding an average performance improvement of +7.82 points.
- **High-Quality Trajectory data Help MLLMs to Plan:** for the 7B models, adding the Tool-Cold-Start phase before ToolGRPO yields a massive performance improvement of +24.93 points on VSP and +19.82 points on Jigsaw compared to using Tool-GRPO alone.

Q: Please guide a player from the start to the goal.

GPT 5: `\boxed{D,D,D,D,D,L,L,L,L,U,U,U,U}` ❌

Ours:

`<think>`The first step is to locate the start point I will use the 'Point' tool.`</think>`
`<tool_call>` Point `</tool_call>`

`<think>`After located the start point, I need to locate the target... `</think>`
`<tool_call>` Point `</tool_call>`

`<think>`After located the start point and target, I need to locate the ice holes ... `</think>`
`<tool_call>` Point `</tool_call>`

`<think>`I will call Astar to find the correct path... `</think>`
`<tool_call>` Astar `</tool_call>`

`<think>`I will call Draw2DPath to verify the path `</think>`
`<tool_call>` Draw2DPath `</tool_call>`

`<think>`Based on observations through tools, I can confirm that the path is correct `</think>`
`<response>`
`\boxed{L,D,L,D,L,D,L,L,L,L,U,U,U}`
`</response>`

Path: LDLD...

✅

(a) VSP-Navigation

Q: which one is the missing part?

A. B.

Ours:

`<think>`I will use the 'DetectBlackArea' tool to find the bounding box of this missing region `</think>`
`<tool_call>` DetectBlackArea `</tool_call>`

`<think>`I will now attempt to insert the first candidate image... `</think>`
`<tool_call>` InsertImage `</tool_call>`

`<think>`does not align seamlessly ... I will now attempt to insert 'img_3' (Option B) into .. `</think>`
`<tool_call>` InsertImage `</tool_call>`

`<think>`I have now inserted both images into the missing region. Upon visual inspection of the resulting images, **neither insertion perfectly resolves the jigsaw puzzle, as the tool's output ... However, I must still determine the most suitable option based on visual evidence ...** 'img_2' (Option A) provides a more plausible and natural completion of the original image. `</think>`
`<response>`
`\boxed{A}` `</response>`

BBOX: [210, 136, 426, 284]

✅

(b) Jigsaw

Q: How can the user access the buying option from the webpage?

GPT5: `\boxed{Click the green "Buy Now >>" button. You can use either of the two on the page:- Top button: [560, 214, 654, 248]}` ❌

Ours:

`<think>` I can see a distinct section on the page that features a book cover and a prominent green button... To accurately describe how a user would access this, I should focus on this specific area of the page.. `</think>`
`<tool_call>` Crop `</tool_call>`

`<think>`To understand how a user would access this buying option, I need to analyze the text ... Now, I will use the OCR tool... `</think>`
`<tool_call>` OCR `</tool_call>`

`<think>`I have successfully used the OCR tool on the cropped image. The button is situated directly beneath the book's title and description, making it a direct and intuitive way for a user to access the purchasing interface. `</think>`
`<response>`
 To access this buying option, a user would likely need to be logged in ... Once logged in, the user would click on the "Buy Now" button ... `</response>`

Buy Now >

✅

(b) GUIQA

Figure 5. Qualitive Results of AdaReasoner-7B.